Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas

Eureka Journal of Computing Science & Digital Innovation (EJCSDI)

ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

EVALUATION OF AUTOMATED PENETRATION TESTING TOOLS FOR DETECTING WEB APP VULNERABILITIES BASED ON OWASP BENCHMARK

Nibras. A. Alkhaykanee
Open Educational College, Al-Qadisiya Center, Al-Qadisiya, Iraq
nbrasa85@gmail.com

Abstract

Web applications are currently among the most often used methods by businesses to engage with their clientele and offer their services. Those applications ought to be safe and compliant with security standards. It is the responsibility of penetration testers to ensure that there are no vulnerabilities that an attacker may exploit, delete, or expose data on the Internet. Thus, the most effective and straightforward method of web application penetration testing is to use automated vulnerability assessment tools; yet, there are advantages and disadvantages to employing these tools. Thus, using the incorrect tool could result in known, anticipated, or undetected vulnerabilities that could allow intrusions. In this study, we assess automated web penetration testing tools using the OWASP Benchmark for vulnerabilities. As an improvement for web penetration testers and penetration testers in the real world, this research employed comparative analysis of penetration testing tools for discovering web app vulnerabilities to assist the process of choosing the right tools based on penetration tester specifications. we performed two scanners, the results showed that OWASP ZAP scored the higher results than Wapiti3. The total amount of vulnerabilities that ZAP found is (11 high, 7 medium, 5 low level and 10 informational). OWASP ZAP covered the next categories: 54% command injection, 64% insecure cookie,11% path

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

traversal, 52% sql injection, 69% cross-site scripting (XSS) and average score of this tool is 23%. Wapiti3 covered only following categories: 11% path traversal, 56% sql injection, 56% cross-site scripting (XSS) and 11% is average score of this tool.

Keywords: Web-application penetration testing; vulnerability detection scanners; penetration testing tools; web application scanners.

1- Introduction

Web apps are become a necessary component of everyday life, yet many of them include significant security vulnerabilities that could be exploited to disastrous effect. Attack strategies are changing in tandem with the advanced technology required to produce these applications [1]. According to Symantec's 2019 Internet Security Threat Report, there was a 56% rise in web based attacks in 2018 and that 30 to 40 million threats were discovered per month on average1. Web application exploitation has been overused against internet based applications in recent years. The technique of simulating cyberattacks against the target system is known as penetration testing. The aim of pen-test is to find vulnerabilities by carefully breaking into a network or web application environment. A penetration tester and a hacker vary in that the former does the test under license, consent, and a signed contract [2]. To find vulnerabilities in software application or web-application design, coding, and configuration of server that result from insecure development techniques we use the web application penetration testing also known as Pen-testing. One of the requirements for using web application penetration testing is to ensure that user testing authentication it does not cause a data breach due to user authentication. To verify secure browser and server configurations must evaluate the web application for security flaws and vulnerabilities like cross-site scripting (XSS) and identifying features that may lead to minimal flaws and guaranteeing web server and database server security

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

[3]. The essential demand for identifying security flaws and vulnerabilities that hackers can take advantage of is using pen-testing [4]. Automated penetration testing tools are several, available and they all have different features and functionalities. It is important to note that the specifications which are available for these automated penetration testing tools are diverse, ranging from scanning a simple one-page web application to scanning a complicated more than one-layer application with multiple workflows at an enterprise level. Benchmarking is a common process used to evaluate the efficiency of web security scanners. Web Input Vector Extractor Teaser is one of the famous benchmarks for web vulnerability scanners (WIVET) [5], the Open Web Application Security Project (OWASP) is a charity committed to enhancing software security [6] like the Web Application Vulnerability Scanner Evaluation Project (WAVSEP) benchmark. Depend on True positive, True Negative, False Positive, and False Negative metrics, utilize the OWASP benchmark for checking the efficacy of two open-source web vulnerability scanners (OWASP ZAP and Wapiti3), is the main goal of this study. These metrics qualify us to provide a comprehensive study of the results and reach out to evaluation for each scanner. The OWASP benchmark utilize these metrics to locate how well a scanner implement.

2- Background

This chapter begins with a study of penetration testing. It outlines penetration testing procedures and discusses a portion of the research works. The remainder of the chapter discusses several automated penetration testing tools and approaches used in this field of study.

2.1 Penetration Testing

Pen test are another name for penetration testing. Penetration testing is a sort of security testing that mimics cyberattacks to identify system or network vulnerabilities before attackers are able to exploit them in the real world, as

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

defined by the National Institute of Standards and Technology (NIST) [7]. A software or hardware flaw in a system that could permit unauthorized use is called a vulnerability [8].Put differently, the aim of penetration testing is to strengthen system security, not to steal or unlawfully access data. It provides highly intuitive insight into existing system security issues for target system admins [9]. Large organizations have been using penetration tests more frequently to safeguard their services and information systems. This enables organizations to address security flaws before they are exploited [10]. Two varieties of penetration testing exist: 1) manual testing and 2) automated testing.

2.1.1 Automated Vs Manual Penetration Testing

The sophistication of cyberattacks is driving an increasing number of businesses to employ security measures. A trillion dollars spent on cybersecurity in the year 2021. It is anticipated that the size of the worldwide PT market will increase from USD 1.7 billion in 2020 to USD 4.5 billion by 2025 [11]. PT is a difficult, costly, and time-consuming process. Furthermore, the proficiency and expertise of a penetration testing team or individual greatly influences the test results. The use of automated PT tools and processes is necessary to increase efficiency. The time, expense, and human labor involved in the collection, processing, and utilization of information can all be greatly decreased with automation.

The degree of automated PT can be divided into four categories [12]:

1- Level 4: Fully autonomous: The system can do any PT task on its own.

2- Level 3: Partially autonomous: The system possesses semi-autonomous capabilities to execute PT tasks. In this instance, human specialists are continuously monitoring the system.

3- Level 2 decision-making assistance mode: the system works in tandem with the human expert to support him or her in making decisions.

4- Learning mode (level 1): While the human tester does PT, the system operates in the background, learning from human decision-makers in the field.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

2.2 Methodology of Penetration Test

There are three fundamental PT kinds:

1- Black box
2- White box
3- Gray box

Black box , this kind of test is also referred to as a "Zero Knowledge" test since the tester is ignorant of the foundational structure and application [13].It depends exclusively upon information gathered from the web app. The majority of automated scanners are built with black box testing capabilities.

White box , the penetration tester gets exposure to all web application details, comprising the server configuration information, user privilege information, code for the application, and infrastructure diagrams. This test, both Both static and dynamic code analyzers are used. The term "full information test" refers to it [13].

Gray box , it combines black box and white box testing. In this case, The infrastructure and the target web-based app are both known to the penetration tester. The tester has application user credentials to enable them to test and find internal vulnerabilities in addition to business logic [13]. When set up properly, automated tools can carry out grey box testing. Grey box testing is not something that an automated tool can undertake on its own; occasionally, manual intervention and setting are needed.

3- Problem statement

Web applications are of great importance in our daily lives because of the crucial role that these applications play in financial and social activities. At the same time, hackers are increasingly exploiting web applications. With the development of the technologies used to exploit these applications, it has become difficult to develop a completely secure web application. However, it is necessary for every organization that deals with sensitive information to ensure the security of this information. Manual testing of vulnerabilities in these applications requires time

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

and effort, in addition to being expensive. While using vulnerability scanning tools for web applications is a solution to the problems of manual testing, So it is necessary to examine The efficiency of the selected tools . These tools must be able to test application vulnerabilities such as Command Injection, Cross-Site Scripting, Insecure cookie, Light Weight Access Protocol (LDAP) Injection, Path Traversal, SQL Injection. This research was undertaken with the aim of compare the automated penetration testing tools by using a framework .We using automated penetration testing tool case studies to explain the frameworks efficiency and usability .Then discuss the statistical investigations of the outcomes obtained from the penetration testing tools.

4- Related Work

The challenge of selecting the appropriate vulnerability detection tools is one that many developers of security- Web services encounter. Current modern tools are not very efficient in terms of vulnerability detection and false positives rates, according to both research and practice. The primary problem is that these tools are meant to be used in very specific environments, which limits the detection approaches they may use. As a result, employing the wrong tools for finding vulnerabilities could lead to the deployment of vulnerability services that are not recognized [14].

In terms of both the quantity of ports found and the tool's discovery time, the authors [7] examined a number of scanning tools. These tools produced a comparative study of the outcomes, which was used to determine which tools were the most effective. After explaining how each step of penetration testing is carried out using the proper tools, the most crucial part of the process is determining a tool's reliability. Therefore, the comparison of four distinct port scanning tools was the main emphasis of this project in order to show how effective they were against the same target. These tools are (Nmap, Dmitry, sparta, unicornscan). The evaluation of port scanning tools reveals that, in terms

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

of identifying the open ports, sparta was the most effective tool. demonstrates that, with the GUI, it was also very user-friendly compared to typing a command in the bash shell, like Dmitry and unicorn scan do.

The authors [15] used Ontology PT and the BDI model to automate PT in real time. In real-world situations, issues like interactivity, dynamic, uncertainty, and complexity can all be handled by the BDI model. Based on SWRL rules for reuse of knowledge and reasoning, the ontology is created for PT. The BDI model can improve its reasoning capacity and determine linkages between targets and vulnerabilities based on Ontology PT. Furthermore, in order to assess IoT security, this thesis has suggested a PT methodology for IoT and its automation that is based on the BDI model. The genuine experiment findings, which include Linux, Windows XP, and Windows 7, show that the BDI model performs better than the manual and other methods now in use. By enlarging the action space and plan, the BDI model may accomplish more intricate and thorough automated attacks on a variety of targets.

The authors [1] evaluate two open source web scanners (ZAP and Arachni) against the OWASP benchmark, they compare the results from the previous benchmark with those of the Web Application Vulnerability Security Evaluation Project (WAVSEP) , a widely used benchmark for assessing the efficacy of scanners. The extent to which each scanner performs differently in a given category is shown by the results of the comprehensive evaluation. For this reason, It cannot be said that there is a comprehensive scanner to detect all vulnerabilities in a web application. But after using two benchmarks to assess two scanner showed that ZAP outperformed Arachni in the SQLI, XSS, and CMDI categories. In contrast, Arachni scored significantly better in the LDAP category. These two scanners' performances varied significantly between the OWASP benchmark and the WAVSEP benchmark. The WAVSEP benchmark showed significantly higher ratings than the OWASP benchmark for all four vulnerability areas and both scanners. This indicate the OWASP benchmark is harder than the

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

WAVSEP benchmark in these four vulnerability categories. It would be better to use the OWASP benchmark as the primary target for evaluating a scanner on these four vulnerability categories, with the WAVSEP benchmark serving as a backup goal to supplement the evaluation findings.

The authors [16] used DVWA and WebGoat, two vulnerable web apps in a study of comparison to determine how well eight web vulnerability scanners (WVSs) could detect vulnerabilities. Among these eight WVSs that were examined, OWASP ZAP, Skipfish, Arachni, Vega, and IronWASP were the five open-source scanners, and Acunitix, HP Webinspect, and IBM Appscan were the three commercial tools. The measures used to evaluate performance are five: the WASSEC, OWASPWBE, Youden index, precision, and recall. The results of the experiment demonstrated that while commercial tools were successful in identifying security flaws, open-source tools like ZIP and Skipfish were also successful in identifying some flaws (such as SQL injection, cross-site scripting, and command execution).

The authors[17]in this study identify potential threats and assess their potential impact by employing the vulnerability assessment and penetration testing (VAPT) technique. This information was then presented to the proprietor of the system via an appropriate framework for engagement that facilitated methodical measurement. For the goal of the research, government websites have been selected in order to illustrate the present trend that took place in cyber community, especially in Indonesia. Numerous vulnerabilities that present 2 (two) serious, 6 (six) medium, and 2 (two) moderate levels of risk have been identified by this investigation. Directory listing, full path disclosure, PHP information disclosure, folder webserver disclosure, and other potential risks are some of these vulnerabilities.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

4- Research Methodology

We talk about the research methods in this part. The processes used in our research approach to compare and assess the chosen web application pen-testing tools are depicted in Figure 1.

Figure 1: Research Methods

3.1 Tools Selection

We have selected two tools (OWASP ZAP and Wapiti3) using their most recent version that have not yet been tested, to the best of our knowledge. These tools were separated into two groups. 1) Scanner tools; 2) Proxy tools. In addition to being used as web proxy, the proxy tools also contain built-in scanner tools [18]. More precise control over the desired request-response interaction is possible with these kinds of technologies. Additionally, it makes it very easy for penetration testers to scan post-login requests. Manual penetration testing is possible for testers due to the proxy component. The scanner tools are stand-alone scanners that are used to do automated scans in Point and Shoot (PaS) setups.

3.1.1 OWASP ZAP

One gratis, open-source pen-test tool is Zed Attack Proxy (ZAP). ZAP is extendable and adaptable, and it was created especially for testing web applications. Fundamentally, ZAP functions as a "man-in-the-middle proxy." It

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

acts as a barrier between the web application and the tester's browser, allowing it to inspect and intercept communications transmitted back and forth between the two, change their contents as necessary, and then forward those packets to their intended location. Many features are offered by ZAP proxy tool, including code review vulnerability detection, web socket, SSL, in-app browser for manual testing, spider tool, active and passive scanners. OWASP ZAP is Graphic user interface (GUI) [19].

Wapiti3

Wapiti permit you audit your websites' or web apps' security. It crawls the deployed web application's webpages in search of scripts and forms where it can inject data, performing "black-box" scans of the application (it does not check the source code). Wapiti operates as a fuzzier once it has the URLs list, forms, and inputs; it injects payloads to check if a script is vulnerable. Wapiti operates as a command-line program [20].

3.2 Creation of an Evaluation Criteria Framework

The first goal of our research is covered in this section. This section explains our comparison matrix framework. We used a framework that is comparable to their approaches after taking into account other web application scanner assessment frameworks, such as [1, 14, 18, 21–23]. However, we employed important parameters to evaluate scanners accurately. We shall go into great detail about the parameters chosen for our matrix. We will be analyzing the web application scanners based on sixteen different parameters. We used a comparative evaluation of tools based on scores. There is a maximum 5-points system for each important parameter as follows: -

1- Tool type: Two categories of tools exist: Graphic user interface (GUI) and Command-line interface (CLI). The majority of pen- tester in PEN-testing web applications preferred the GUI interface rather than CLI. We will also talk about

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

whether the tool is a scanner that interacts directly with the target for scanning, or a web proxy that operates by intercepting a browser request.

2- Penetration Testing method: Current scanning tools are capable of capturing web application sessions and identifying source code variances in web applications. The majority of tools for automated PEN testing of web applications employ the black box test technique throughout authenticated scans. The penetration testing type results :
1: Only employ the black box method.
2: Gray box and black box method.
3: Testing methods include black box, gray box, and white box.

3- Varieties of crawling: One kind of scan is called crawling, which includes map the application request by request. The links that the crawler finds during the process are saved, and then uses those links later on for scanning. Crawling comes in two types: 1) active crawling and 2) passive crawling. In order to obtain the active links, By submitting requests to the server of application, the active crawler communicates with the application. Without actively interacting with the application, a passive crawler operates in silence. The passive crawler operates while using the application manually. The crawling function's score is:
1: either only an active crawler or only a passive one.
2: two types of crawlers: passive and active.

4- URLs number covered: A step in the PEN-testing process called information gathering includes web application crawling. At this stage, a penetration tester aims to learn as much as possible about the web application. Counting URLs number that the scanner crawls can be used to indicate crawler coverage. The score increases as the URLs number covered by the scanner. Score for URLs that are covered:

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

1: Less than 25%.
2: From 25% to 50%.
3: From 50% to 70%.
4: From 70% to 90%.
5: More than 90%.

5-Time of scanning: The penetration tester uses the automated tool to cover a larger area in a huge application. As a result, the time used is significant for the scanners evaluation .Score for time of scanning:
1: At least six hours.
2: Exceeding three hours.
3: Exceeding 45 min.
4: In less half an hour.

6- Types of Scanning: In web application pen testing, there are two kinds of scans: passive and active scan. The scanner that offers both active and passive scan receives the highest rating in this metric. Score for Types of Scanning:
1: either only the active scan or just the passive scan.
2: scanning both actively and passively.
3: passive, policy-scan, or active.

7- Reports' characteristics: A new feature in scanners is the ability to Reports format according to the compliance policy that needs to be examined by the penetration tester. These standards include HIPAA, OWASP Top 10, and others. Reports can be found in several common forms, including HTML, PDF, and XML. The penetration tester can more easily and thoroughly analyze the compliance policy reports. Score for Features of Reports:
0: Reports in HTML, PDF, and XML.
1: Reports of Standards compliance, including HIPAA and the OWASP Top 10.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

8- Added Features: Certain automated tools contain extensions and add-ons that improve the scanner's ability to identify vulnerabilities. The majority of penetration testers utilize these functionalities. Score for features and add-ons of the extension:
0: without any extensions or add-ons.
1: includes features from extensions and add-ons.

9- Configuration simplicity: How simple it is to utilize the scanner and what dependencies it has are determined by its simplicity of configuration. Three configuration levels are defined: 1) Simple (Plug and Play) application that is ready to use right out of the box, 2) Difficult: requiring certain prerequisites, like installing PHP and Java 3) Difficult (Expert level): Requiring particular configuration for the server and database.

10- Logging Option for Scans: In order to monitor and identify thousands of requests and responses during PEN-testing, the logs are necessary. It's important to log these processes so you can access them later. Result for scan logs:
0: No option to log the scan.
1: the option to log the scan.

11- Tool Cost: When choosing a tool, cost is an important factor. Even though numerous tools have identical features and functionalities, there is a significant price difference based on brand. Because of its active development community, several freeware tools also outperform commercial tools in terms of functionality and performance.

12- The capability of pausing and resuming scanning: One advantage of the scanner is that it can be stopped and started again from the same location, which

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

saves the pen-tester time when rescanning the application. Score for this parameter:
0: No capability of pausing and resuming scanning
1: The capability of pausing and resuming scanning.

13- Coverage of the OWASP Top 10 Vulnerabilities: Penetration testers protect their assets from vulnerabilities by using penetration testing tools to cover the top ten vulnerabilities in their apps. The OWASP Top 10 Vulnerabilities are important for assessing the process. Avoiding these top 10 vulnerabilities is also an objective for software testers and developers. Based of all the vulnerabilities that are currently present in the OWASP benchmark, this parameter will assess the level of covered vulnerabilities. Score for coverage of vulnerabilities:
1: Under 25%
2: From 25% to 50%
3: From 50% to 70%
4: Over 70%

14- Reports of false positives: Vulnerabilities that the scanner listed as positives but did not exist in the application are known as false positives. The scanner with a lower rate results of false positive is better.

$$FPR = \frac{FP}{FP+TN} \times 100 \quad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

The false positive number's score is:
1: More than 50%
2: More than 30%
3: Below 30%

15-Reports of True Positives: The term "true positive" refers to the scanner's accurate detection of the true vulnerability identification in the OWASP benchmark.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

$$TPR = \frac{TP}{TP+FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

The True positive number's score is:

1: less than 10%
2: Reaching 20%
3: Reaching 50%
4: More than 50%

16-Accuracy Score of the OWASP Benchmark: A traditional method of quantifying the accuracy of a series of tests is the Youden Index, which is essentially what the Benchmark Accuracy Score is. Youden's index is one of the earliest metrics for evaluating accuracy. Youden's index is computed by subtracting 1 from the total of a test's reported sensitivity and specificity, which are expressed as a fraction of a whole number rather than as a percentage: (specificity plus sensitivity) - 1. In the case of a test with low diagnostic accuracy, Youden's index is equal to 0 and in the case of a perfect test it is equal to 1[24]. Youden index is equal to [25]

$$\frac{TP}{TP+FN} + \frac{TN}{TN+FP} - 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

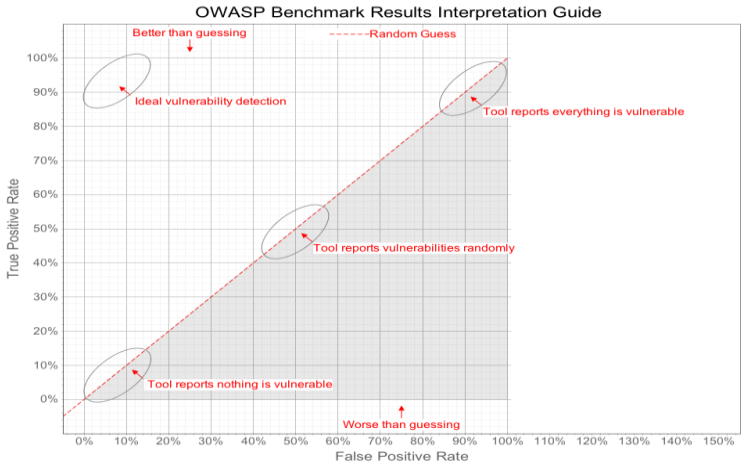


Figure 2: An explanation guide for OWASP

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

In figure 2 the line length from the point on the graph down to the diagonal "guessing" line represents the Benchmark Score. Remember that a benchmark score could really be negative if a point is below the line. When the True Positive Rate truly is lower than the False Positive Rate, this happens.

5- Implementation

We will talk about the artifact implementation in this part. To assess the tool's scanning, crawling, and capacities for detecting vulnerabilities we have used the OWASP benchmark test tool [26], we first set up the environment before configuring the scan settings and beginning the benchmarking process to assess the coverage of vulnerability detection and crawling by the scanner. Following benchmarking, we used our evaluation approach to compare and analyze the results.

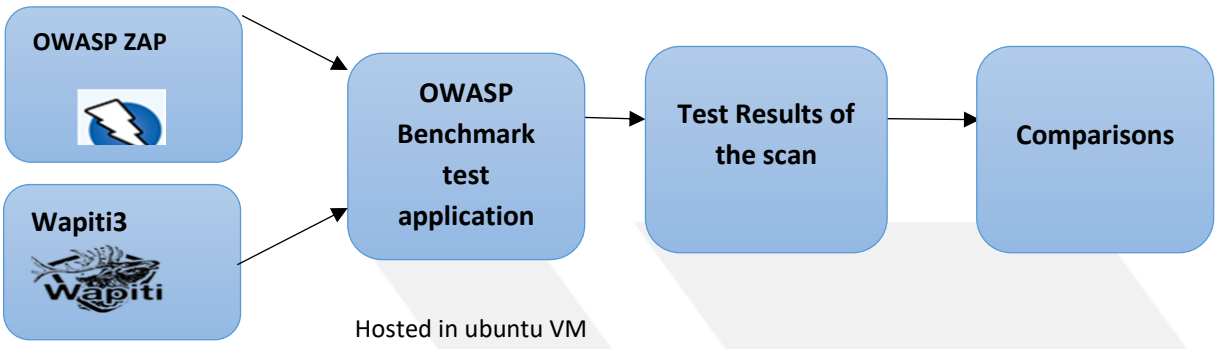


Figure 3: Process of evaluating scanners

4.1 OWASP Benchmark test:

The OWASP benchmark test is a free and open-source vulnerability test case set built on Java that is intended to assess the speed, crawler coverage, and accuracy of automated scanners' vulnerability detection. This cutting-edge benchmarking tool is updated frequently. It contains more than 2740 examples of vulnerabilities from the top 10 OWASP categories. Up till now, it hasn't been utilized to assess

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

the most recent versions of OWASP ZAP, wapiti3. Through the browser, go to https://localhost:8443/benchmark/ to visit the application.

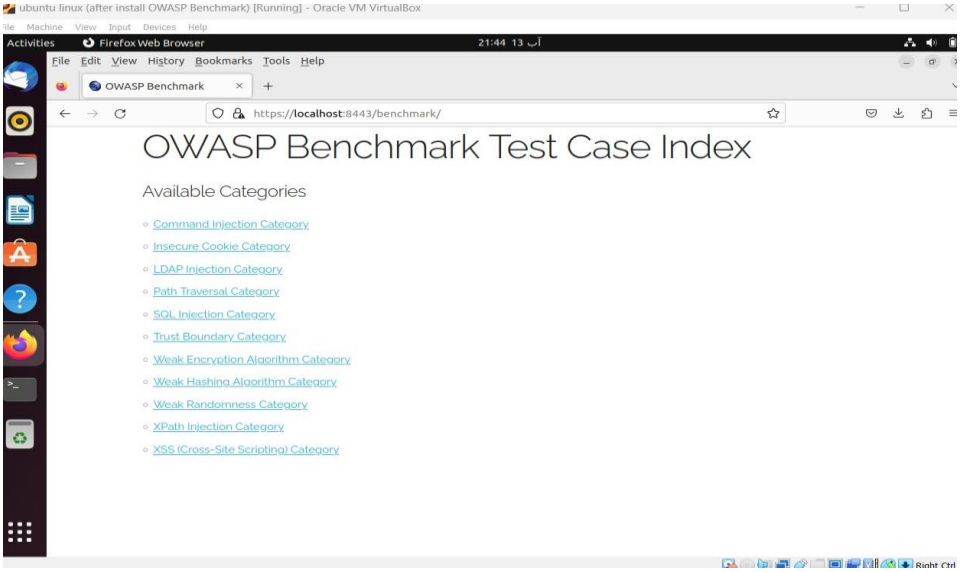


Figure 4: The application OWASP Benchmark

5.2 Setting Up the Environment

The following setup is part of the evaluation procedure. Setup is configured on the Windows host computer and Ubuntu Linux virtual machine:

Step 1: Installing the scanner (OWASP ZAP, wapiti3) in order to attack the OWASP benchmark application is part of setting up the environment. The installation of services and apps that were dependent on it was also included. In order to use OWASP ZAP, we first set up the browser proxy and use it to intercept the application. Next, request by request, we manually crawled the application. Next, using the tool, we actively crawled the application. Additional URLs were found in the application and placed under the target scope. We updated the tool's scope section with the target URL. Only the target URL will be intercepted and scanned by the tool. Next, we click ZAP's "Active scan" button to begin the scan on the benchmark target system.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

To conduct the scan for wapiti3, we use the command line interface. According to the configuration manual's instructions, we install Wapiti.
Using the command "wapiti" and the application root URL, we carry out the point-and-shoot scan. Additionally, we modify the command to save the scan results in our project folder by adding switches.
Step 2: Each tool's ".xml" output is required in order to obtain the benchmark evaluation. We default to using the ".xml" format for the scan output for OWASP ZAP and Wapiti.
Step 3: The tool results are analyzed by the OWASP benchmark, which generates compliance output. Every scanner's XML result file was copied to the 'results' folder inside the 'benchmark' root folder. To generate the benchmark results, execute the "createScorecard.sh" command. For assessment, we have mostly focused on the five vulnerabilities listed in the OWASP benchmark. Command injection, XSS, SQL injection, Path Traversal, and insecure cookies are among these ten vulnerabilities [25]. In addition to these ten vulnerabilities, we take into account additional vulnerabilities that the scanners reported in order to assess the scanning tool as a whole.
In the part that follows, we've talked about the benchmark test results and other framework parameters.

6. Experimental Results

Our results from executing an experimental scan on the OWASP benchmark test application are presented in this chapter. Additionally, we ranked the scanners by comparing them using our framework and calculating test scores for each scanner.

1) Tool type :

The web proxy tool, OWASP ZAP, is built on a GUI. Any user level can easily use GUI-based tools. We discovered during our trial that ZAP was more user-

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

friendly because all of the modules were readily available and had clear instructions. One feature of OWASP ZAP is the capability to launch a browser directly from the tool. This browser comes pre-configured with ZAP proxy, so the user doesn't need to manually prepared it.
Conversely, Wapiti3 uses only a command line interface (CLI) and it's a scanner that interacts directly with the target.

2) Penetration Testing method:
Black box testing can be done by OWASP ZAP without the need for manual intervention. Gray box test this scanner has add-ons that can manage login requests and credentials. This can be managed in OWASP ZAP by utilizing the "session properties" settings. The session can continue to run while the scanning procedure is underway thanks to these add-ons' capacity to handle login-logout requests. These tools execute the above two test types as well as the Dynamic Security Application Test (DAST) type of test. Wapiti3 is using black box only.

3) Varieties of crawling:
OWASP ZAP have active and passive scanning feature which allows application crawl logging while browsing the application. While wapiti3 able to crawl actively only.

4) URLs number covered:
The OWASP benchmark contains approximately 5500 URLs, including the pages and activities (login, submit, etc.), ZAP was able to find 5507 URLs in the OWASP benchmark project. 5507 URLs and Wapiti3 was able to attacked 10,327 pages for module csp ,module cookieflags , module xss ,module file, module sql , module upload , module ssrf , module http_headers, 94 pages for module exec and 8359 pages for module redirect. Which mean both tools covered all the benchmark URLs.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

5) Time of scanning:
A scan with OWASP ZAP took three hours and thirty minutes and Wapiti3 scan time was over 10 hours .It was not possible for either scanner to receive a high score in this category. This could be because the benchmark is a big, completely vulnerable application.

6) Types of Scanning:
OWASP ZAP outperformed Wapiti3 in types of scanning. Because OWASP ZAP can operate in both active and passive scan options, it scored the maximum three points, also in ZAP you can define as many scan policies as you like and you can define the default scan policy to be used for active scans and for the attack mode. Wapiti3 received 2 points because it's ability to use both active and passive scan modes.

7) Reports' characteristics: In this criteria, OWASP ZAP and Wapiti3 received a score of 0, resulting in their expected standard reports. They don't have any more features for reporting

8) Added Features: When it came to adding extensions and add-ons for a stronger scanner and better vulnerability detection, OWASP ZAP scored higher points than Wapiti3. Also, it has features an upgraded marketplace where you can install add-ons. Wapiti3 received a score of 0 since they were unable to add to this scanner extensions and add-ons.

9) Configuration simplicity: We observed that both tools fall under the category of being simple to configure and use. OWASP ZAP and Wapiti3 needed less requirements, such as JAVA for OWASP ZAP and Python 3 for Wapiti3.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

10- Logging Option for Scans: Based on the ability to log scans, OWASP ZAP scored 1 point, which is higher than Wapiti3.While Wapiti3 received a score of zero due to the lack of a logging scans, OWASP ZAP was able to log every request and response received during the scan.

11) Cost of the tool:
Each tool Wapiti3 and OWASP ZAP are free. To remain up to date with the latest vulnerability detection techniques, WASP ZAP is updated more quickly than the others, which is an excellent feature.

12- The capability of pausing and resuming scanning: Both Wapiti3 and OWASP ZAP have the ability to stop and resume processes. Wapiti3 and OWASP ZAP both received one point.

13- Coverage of the OWASP Top 10 Vulnerabilities:
The total amount of vulnerabilities that ZAP found is (11 high, 7 medium, 5 low level and 10 informational). When it comes to covering the top 10 vulnerabilities in one scan, OWASP ZAP outperforms Wapiti3. In one scan OWASP ZAP covered the next categories: 54% command injection, 64% insecure cookie,11% path traversal, 52% sql injection, 69% cross-site scripting (XSS) and average score of this tool is 23%. Wapiti3 covered only following categories: 11% path traversal, 56% sql injection, 56% cross-site scripting (XSS) and 11% is average score of this tool.

14) Reports of false positives: Based on benchmark outcomes. OWASP ZAP reported 1.21% while Wapiti3 was 0.00%. Both tools scored 3 points.

15) Reports of True Positives: OWASP ZAP was able to detect 23.97% and Wapiti3 11.20% of the vulnerabilities in the application. Then OWASP ZAP scored 2 points and Wapiti3 scored 1 point.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

16) Accuracy Score of the OWASP Benchmark: Youden Index of OWASP ZAP is 0.2975 and Wapiti3 is 0.2155.

No.	Parameters	OWASP ZAP	Wapiti3
1	Tool type	GUI Proxy	CLI scanner
2	Penetration Testing method	2	1
3	Varieties of crawling	2	1
4	URLs number covered	5	5
5	Time of scanning	2	1
6	Types of scanning	3	2
7	Reports' characteristics	0	0
8	Configuration simplicity	1	1
9	Cost of the tool	Free	Free
10	Logging Option for Scans	1	0
11	The capability of pausing and resuming scanning	1	1
12	Coverage of the OWASP Top 10 Vulnerabilities	1	1
13	Reports of false positives	3	3
14	Reports of True Positives	2	1
	Total Points	23/34	17/34

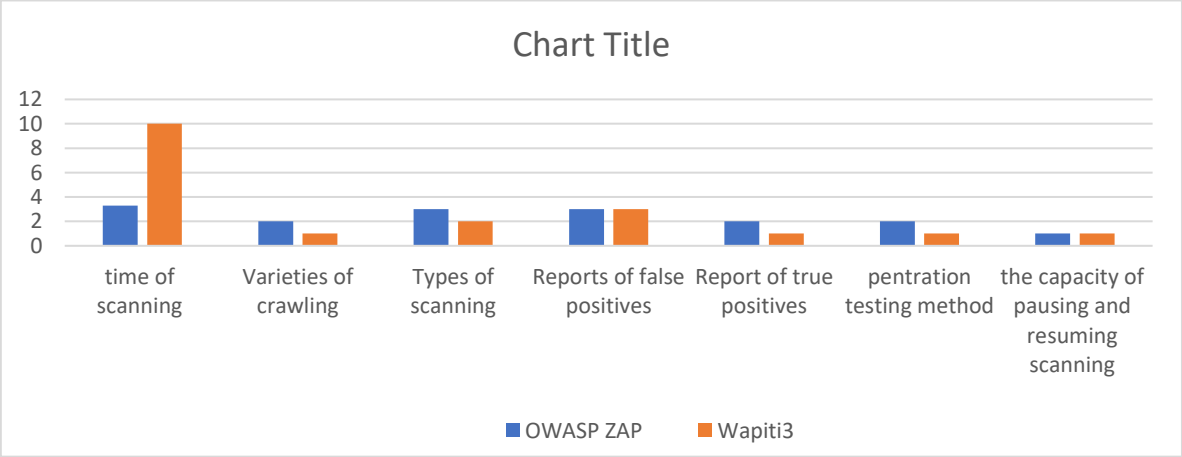
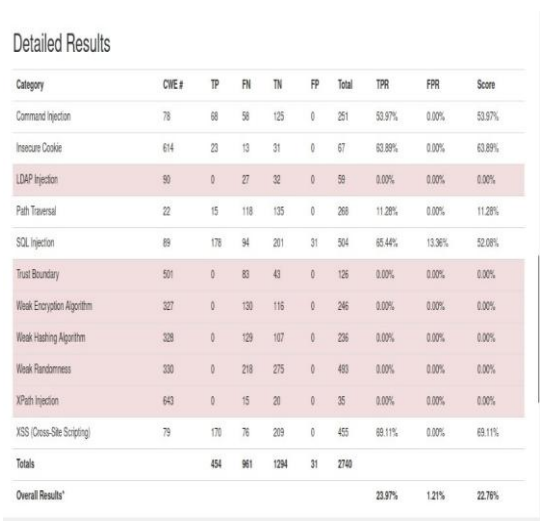


Figure 5: ZAP and Wapiti3 score comparison by metric

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

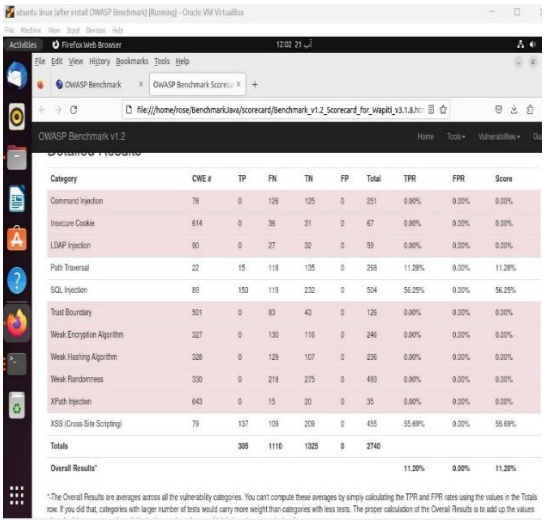
https://eurekaoa.com/index.php/10



Detailed Results

Category	CWE #	TP	FN	TN	FP	Total	TPR	FPR	Score
Command Injection	78	68	58	125	0	251	53.97%	0.00%	53.97%
Insecure Cookie	614	23	13	31	0	67	63.89%	0.00%	63.89%
LDAP Injection	90	0	27	32	0	59	0.00%	0.00%	0.00%
Path Traversal	22	15	118	135	0	268	11.28%	0.00%	11.28%
SQL Injection	89	178	94	201	31	504	65.44%	13.36%	52.08%
Trust Boundary	501	0	83	43	0	126	0.00%	0.00%	0.00%
Weak Encryption Algorithm	327	0	130	116	0	246	0.00%	0.00%	0.00%
Weak Hashing Algorithm	328	0	129	107	0	236	0.00%	0.00%	0.00%
Weak Randomness	330	0	218	275	0	493	0.00%	0.00%	0.00%
XPath Injection	643	0	15	20	0	35	0.00%	0.00%	0.00%
XSS (Cross-Site Scripting)	79	170	76	209	0	455	69.11%	0.00%	69.11%
Totals		454	961	1294	31	2740			
Overall Results*							23.97%	1.21%	22.76%

Figure 5: OWASP ZAP Results



OWASP Benchmark v1.2

Category	CWE #	TP	FN	TN	FP	Total	TPR	FPR	Score
Command Injection	78	0	126	125	0	251	0.00%	0.00%	0.00%
Insecure Cookie	614	0	36	31	0	67	0.00%	0.00%	0.00%
LDAP Injection	90	0	27	32	0	59	0.00%	0.00%	0.00%
Path Traversal	22	15	118	135	0	268	11.28%	0.00%	11.28%
SQL Injection	89	153	119	232	0	504	56.25%	0.00%	56.25%
Trust Boundary	501	0	83	43	0	126	0.00%	0.00%	0.00%
Weak Encryption Algorithm	327	0	130	116	0	246	0.00%	0.00%	0.00%
Weak Hashing Algorithm	328	0	129	107	0	236	0.00%	0.00%	0.00%
Weak Randomness	330	0	218	275	0	493	0.00%	0.00%	0.00%
XPath Injection	643	0	15	20	0	35	0.00%	0.00%	0.00%
XSS (Cross-Site Scripting)	79	137	109	209	0	455	55.69%	0.00%	55.69%
Totals		305	1110	1325	0	2740			
Overall Results*							11.20%	0.00%	11.20%

Figure 6: Wapiti3 Results

7. Conclusions

We presented a comparative evaluation of the two Pen-testing tools for web applications (OWASP ZAP, Wapiti3) utilizing our benchmarking framework. While every tool has advantages and disadvantages, the web proxy tool proved to be more effective when compared to the benchmark application. ZAP outperformed Wapiti3 across a range of areas, ZAP was superior in command injection, insecure cookie and cross-site scripting (XSS). However, Wapiti3 fared significantly better in SQL injection. OWASP ZAP received the highest score in the experiments, according to our comparison framework assessment. The test revealed that no single scanner is suitable for all types of vulnerabilities. Every scanner operates at a different detection rate. As a result, the right scanner should be employed to find a specific vulnerability. It is preferable to have more than one scanner available for dynamic web app penetration testing. Not every vulnerability is reported by every scanner. We assessed two tools in this study using the OWASP Benchmark application. Based on this application and its build the vulnerabilities discovered or overlooked. However, this research can be carried out and further assessed other variety of benchmarking applications.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas

Eureka Journal of Computing Science & Digital Innovation (EJCSDI)

ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

References

[1] Mburano, Balume, and Weisheng Si. "Evaluation of web vulnerability scanners based on owasp benchmark." *2018 26th International Conference on Systems Engineering (ICSEng)*. IEEE, 2018.

[2] Dalalana Bertoglio, Daniel, and Avelino Francisco Zorzo. "Overview and open issues on penetration test." *Journal of the Brazilian Computer Society* 23.1 (2017): 1-16.

[3] Im, J.; Yoon, J.; Jin, M. Interaction Platform for Improving Detection Capability of Dynamic Application Security Testing. In Proceedings of the 14th International Joint Conference on e-Business and Telecommunications, Madrid, Spain, 26–28 July 2017;
pp. 474–479.

[4] Li, J. Vulnerabilities mapping based on OWASP-SANS: A survey for static application security testing (SAST). Ann. Emerg. Technol. Comput. (AETiC) **2020**, 4, 1–8.

[5] E. T. İslam and B. Urgun, "WIVET—Benchmarking Coverage Qualities of Web Crawlers," *The Computer Journal,* vol. 60, no. 4, p. 555–572, March 2017.

[6] S. Chen, "WAVSEP – Web Application Vulnerability Scanner Evaluation Project," 10 November 2017. [Online]. Available: http://sectooladdict.blogspot.com/. [Accessed 12 November 2019].

[7] Mamilla, Sushmitha Reddy. "A Study of Penetration Testing Processes and Tools." (2021).

[8] P. Xiong and P. Liam, "A Model-Driven Penetration Test Framework for Web applications," in Eighth Annual International Conference on Privacy, Security and Trust, Ottawa, ON, Canada, 2010.

[9] Matthew Denis, Carlos Zena, and Thaier Hayajneh. Penetration testing: Concepts, attack methods, and defense strategies. In 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), pages 1{6. IEEE, 2016.

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

[10] Morgan, Steve. "Global cybersecurity spending predicted to exceed $1 trillion from 2017-2021." *Cybercrime Magazine* 10 (2019).

[11] Steve Morgan. Cybersecurity talent crunch to create 3.5 million unfilled jobs globally
by 2021. Cybercrime Magazine, 2019.

[12] Mohamed C Ghanem and Thomas M Chen. Reinforcement learning for intelligent penetration testing. In 2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), pages 185{192. IEEE, 2018.

[[13] K. Shaukat, A. Faisal, R. Masood, A. Usman and U. Shaukat, "Security quality assurance through penetration testing," in *19th International Multi-Topic Conference (INMIC)*, Islamabad, Pakistan, 2016.

[[14] Albahar, Marwan, Dhoha Alansari, and Anca Jurcut. "An empirical comparison of pen-testing tools for detecting web app vulnerabilities." *Electronics* 11.19 (2022): 2991.

[15] Chu, Ge. *Automation of Penetration Testing*. The University of Liverpool (United Kingdom), 2021.

[[16] Amankwah, Richard, et al. "An empirical comparison of commercial and open-source web vulnerability scanners." *Software: Practice and Experience* 50.9 (2020): 1842-1857.

[[17] Almaarif, Ahmad, and Muharman Lubis. "Vulnerability Assessment and Penetration Testing (VAPT) Framework: Case Study of Government's Website." *International Journal on Advanced Science Engineering and Information Technology* 10.5 (2020): 1874-1880.

[[18] Shah, Mandar Prashant. Comparative analysis of the automated penetration testing tools. Diss. Dublin, National College of Ireland, 2020.

[[19] OWASP. *OWASP Zed Attack Proxy Project*. Available: https://www.zaproxy.org/getting-started/#security-testing-basics .

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

[20] Wapiti 3.1.5 Available: https://wapiti-scanner.github.io/ .
[21] Matti, Erik. "Evaluation of open source web vulnerability scanners and their techniques used to find SQL injection and cross-site scripting vulnerabilities." (2021).
[22] Almaarif, Ahmad, and Muharman Lubis. "Vulnerability Assessment and Penetration Testing (VAPT) Framework: Case Study of Government's Website." *International Journal on Advanced Science Engineering and Information Technology* 10.5 (2020): 1874-1880.
[23] Wakhale, Ajinkya. "Web Application Vulnerability Assessment Tools Analysis." *UMBC Student Collection* (2018).
[24] https://owasp.org/www-project-benchmark/ .
[25] Youden, W.J. *(1950).* "Index for rating diagnostic tests". *Cancer. 3: 32–35.* doi:10.1002/1097-0142(1950)3:1<32::aid-cncr2820030106>3.0.co;2-3. PMID 15405679.
[26] D. Wichers, "OWASP Benchmark," OWASP, 05 June 2016. [Online]. Available: https://www.owasp.org/index.php/Benchmark#tab=Main.

Web applications are of great importance in our daily lives because of the crucial role that these applications play in financial and social activities. At the same time, hackers are increasingly exploiting web applications. With the development of the technologies used to exploit these applications, it has become difficult to develop a completely secure web application. However, it is necessary for every organization that deals with sensitive information to ensure the security of this information. Manual testing of vulnerabilities in these applications requires time and effort, in addition to being expensive. While using vulnerability scanning tools for web applications is a solution to the problems of manual testing, it is necessary to examine The efficiency of the selected tools in some functions such as web crawling and spamming. These tools must be able to test application vulnerabilities such as Command Injection, Cross-Site

Eureka Open Access Journals
Open Access | Peer Reviewed | International Journals | Multidisciplinary Areas
Eureka Journal of Computing Science & Digital Innovation (EJCSDI)
ISSN 2760-4993 (Online) Volume 2, Issue 6, June 2026

https://eurekaoa.com/index.php/10

Scripting, Insecure cookie, Light Weight Access Protocol (LDAP) Injection, Path Traversal, SQL Injection.

This research was conducted with an objective of developing a framework to compare the automated penetration testing tools. We demonstrate the frameworks efficiency and usability using automated penetration testing tool case studies. The research framework is based on the previous research done by the Mayur Turuvekere and Anala A. Pandit proposing automated penetration testing tools evaluation based on vulnerabilities identified [4]. This research widens the evaluation matrix and provides a broader scale which includes not only vulnerability detection, but also other parameters of the penetration testing tools. The research is performed using statistical investigations of the outcomes obtained from the penetration testing tools.

Research Question:

• To identify efficient automated penetration testing tool to suffice the current day industry requirement.

Objective:

• Develop Framework to compare the web applications penetration testing tools.

• To do the research based comparative analysis of automated penetration testing tools on recent trends in the industry.

• To demonstrate statistical investigations of the outcomes obtained from the penetration testing tools.